Probabilistic Self-Localization for Mobile Robots

Clark F. Olson

Jet Propulsion Laboratory, California Institute of Technology

4800 Oak Grove Drive, Mail Stop 125-209, Pasadena, CA 91109-8099

http://robotics.jpl.nasa.gov/people/olson/homepage.html

Abstract

Localization is a critical issue in mobile robotics. If the robot does not know where it is, it cannot effectively plan movements, locate objects, or reach goals. In this paper, we describe probabilistic self-localization techniques for mobile robots that are based on the principal of maximum-likelihood estimation. The basic method is to compare a map generated at the current robot position to a previously generated map of the environment to probabilistically maximize the agreement between the maps. This method is able to operate in both indoor and outdoor environments using either discrete features or an occupancy grid to represent the world map. The map may be generated using any method to detect features in the robot's surroundings, including vision, sonar, and laser range-finder. A global search of the pose space is performed that guarantees that the best position in a discretized pose space is found according to the probabilistic map agreement measure. In addition, fitting the likelihood function with a parameterized surface allows both subpixel localization and uncertainty estimation to be performed. The application of these techniques in several experiments is described, including experimental localization results for the Sojourner Mars rover.

1 Introduction

In order to navigate effectively in the environment and interact with the surroundings, mobile robots must have some method by which to determine their position with respect to known locations in the environment. This is called the *localization problem.* The most common and basic method for performing localization is through dead-reckoning. This technique integrates the velocity history of the robot over time to determine the change in position for the starting location (see, for example, [5, 39]). Unfortunately, pure dead-reckoning methods are prone to errors that grow without bound over time, so some other method is necessary to periodically update the robot position in order to correct this error.

It is common to combine some additional localization technique such as triangulation from landmarks or map matching with dead-reckoning using an extended Kalman filter to probabilistically update the robot position. In this paper, we describe a technique that

performs localization infrequently to update the position of the robot. In order to perform localization, we compare a map generated using the robot's sensors at the current position (the *local map*) to a previously generated map of the environment (the *global map*) (which may be constructed as the robot explores). The maps are matched according to a maximum-likelihood similarity measure. The best relative position between the maps is found using branch-and-bound search techniques that do not require an initial estimate of the robot position to yield good results, only bounds on the search space, which may be of any size. This technique is general enough to use with either a discrete, landmark-based map representation or an occupancy grid model of the environment. We have primarily explored the application of these techniques to three-dimensional occupancy grids in order to model unstructured outdoor terrain.

The measure that we use to compare the maps is derived from previous work on image matching using the Hausdorff distance [20]. We have reformulated this measure in terms of maximum-likelihood estimation. In this measure, the likelihood of each position is formulated as the product of the probability distribution functions of the distances from the features in the local map to the closest features in the global map, with an additional term representing the prior probability of the position. This probabilistic measure avoids the drawbacks of the original matching measure, which include an sharp boundary between good and poor feature matches and the inability to incorporate probabilistic information, while retaining the advantages, which include robustness to outliers and a global search technique [30]. This measure also allows for both subpixel localization in discretized pose spaces and accurate estimation of the uncertainty in the localization by fitting the likelihood function with a parameterized surface.

The strategy that we use to locate the best position is a hierarchical divide-and-conquer algorithm over the space of possible model positions (the *pose space*) that has been recently used for matching image edge maps [21, 31, 34]. We first test the position given by dead-reckoning so that we have an initial position to compare against. The pose space is then divided into rectilinear cells. For each cell in the space, we attempt to prove that the cell cannot contain a position that is superior to the best one that has been found so far using an efficient bounding mechanism. For any cell that cannot be pruned, the cell is divided into smaller cells and the process is repeated recursively. The process stops dividing the cells when they have become small enough to represent valid hypotheses or by some other robust stopping criterion. This is implemented using a depth-first search of the cell hierarchy.

Our motivation for this work is the Long Range Science Rover project at JPL, which has developed the Rocky 7 Mars rover prototype [18]. Mars rovers need increased self-localization ability in order to perform with greater autonomy from both operators on Earth and from the lander bringing the rover to Mars. For example, the Sojourner Mars rover was limited to moving short distances during each downlink cycle due to positional uncertainty and could not venture far from the lander. The method by which dead-reckoning errors were corrected for Sojourner was through a human operator overlaying a model of the rover on stereo range data that was computed from downlinked imagery of the rover taken by the lander [35].

The techniques described here are effective whenever a dense range map can be generated

in the robot's local coordinate frame and we have a range map of the same terrain in the frame of reference in which we wish to localize the robot. Alternatively, we could use discrete landmarks such as rock positions to perform localization. We can thus use rover imagery, either from the close-to-the-ground navigation cameras or from a rover mast such as the one on Rocky 7 (see Figure 1) to generate the local map. The global map might also be created from the rover mast or navigation imagery, but it could also be generated from imagery from the lander (including descent imagery), and it is possible that orbital imagery could be used, although we will not have orbital imagery of sufficient resolution to use for rover localization with sub-meter accuracy in the near future [27].

The localization techniques described here are very useful in the context of a Mars mission. While operating in a small area containing several science targets (such as the area around the lander that Sojourner operated in), we may perform localization using the panoramic imagery generated at the center of the area as our global map. While this is not crucial when the lander can see the rover, the next-generation Mars rover will venture away from the lander and it will be equipped with a mast with stereo cameras that will allow it to generate panoramic imagery of the terrain. This allows localization to be performed by matching the panoramic range maps generated using the mast imagery to maps generated from either the navigation cameras, if possible, or by using the mast to image interesting terrain, if necessary. These techniques can also be used on traverses between sites by performing localization at some interval in order to update the position of the rover.

We have tested these techniques using real and synthetic data. The synthetic experiments model a case where the robot performs localization using a discrete set of known landmarks in the environment. These experiments indicate that accurate localization can be achieved by searching a discretized pose space through the use of subpixel estimation and that the uncertainty in the localization can be accurately modeled by fitting the surface of the likelihood surface. Our application of these techniques to real data creates an occupancy grid representation of the terrain using stereo vision [25], since we are concerned primarily with performing localization in natural terrain. Experiments matching such range maps from stereo vision have been performed with both terrestrial data, acquired in the JPL Mars Yard using the Rocky 7 research rover [18], and imagery of Mars acquired by the Mars Pathfinder lander and Sojourner rover [35]. The experiments using Mars imagery validate the use of these techniques to perform autonomous localization for Mars rovers without the need to downlink information to Earth.

In the following section, we review previous work on robot localization, focusing on techniques that perform some variation of map matching to locate the robot. Section 3 describes the probabilistic map similarity measure that we use to determine which position of the robot is the most likely to be correct. Section 4 gives an algorithm for searching the space of possible robot positions to locate the position that maximizes this map similarity measure. This section also discusses the application of these techniques to robot localization from discrete landmarks and by matching occupancy maps. The techniques by which we perform subpixel estimation in discretized pose spaces and estimate the uncertainty in the localization process are given in Section 5. The experimental results that we have achieved

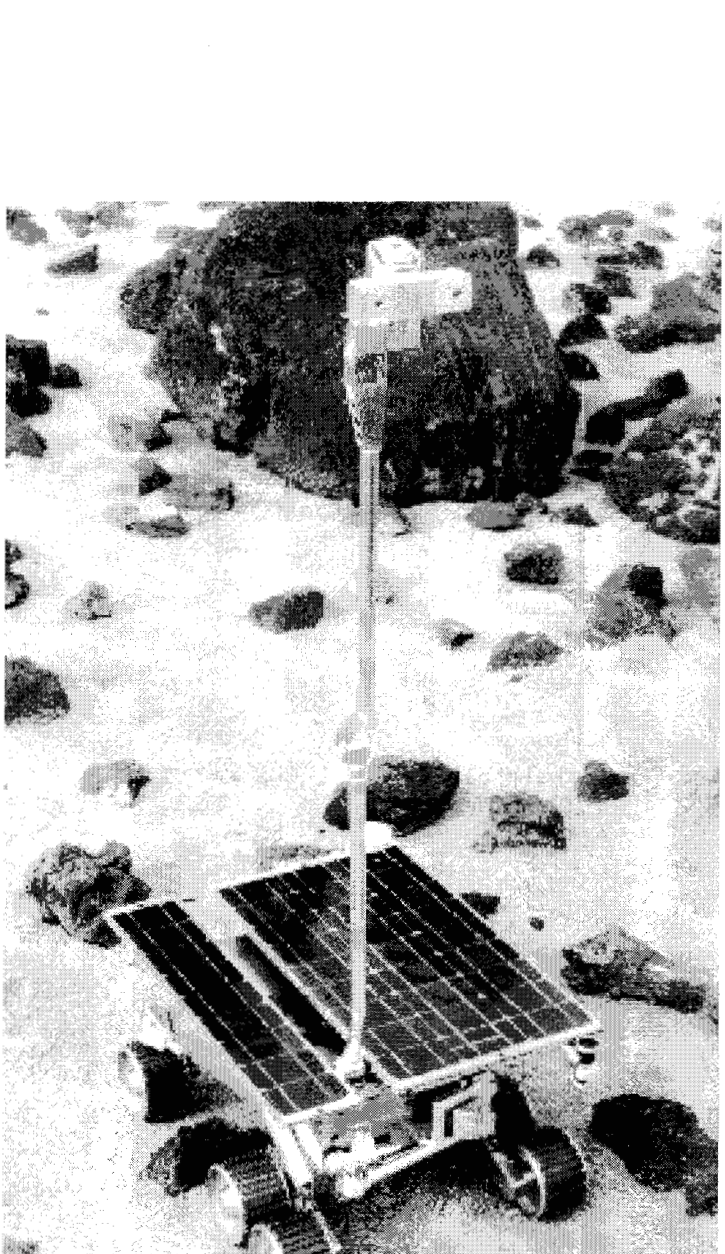


Figure 1: Rocky 7 Mars rover prototype in the JPL Mars yard with mast deployed.

with real and synthetic data are described in Section 6. Finally, Section 7 summarizes the paper and gives some concluding remarks. We note that portions of this work have been presented at recent conferences [29, 30, 32].

2 Previous work

Mobile robot localization is typically performed by combining the results of dead-reckoning and some periodic sensor-based localization technique, for example with an extended Kalman filter. Many techniques have been used to provide the periodic sensor-based localization. Often these techniques operate by determining correspondence between a set of sensed features (such as landmarks) and a known map of the environment. The known positions of the features in the map, together with the sensed positions relative to the robot allow the robot's current position to be inferred. We give a brief review of several such techniques here.

In some cases, the sensor-based techniques operate frequently, so that the robot moves a small amount between localization steps. This simplifies the problem, because the determination of the feature correspondences can be treated as a tracking problem, rather than searching the map for the features. The drawback to this formulation is that the techniques must operate frequently enough to prevent the tracker from losing track of the features. If the tracker makes a mistake by specifying an incorrect correspondence, it may have a drastic effect on the localization result. Examples of techniques based on the Kalman filter include [2, 10, 19, 23, 43]

Many other methods have been proposed that do not require frequent sensor measurements. One such method is to locate nearby landmarks and to perform a triangulation procedure to determine the position of the robot. Sugihara addressed the problem where the relative direction of the landmarks can be sensed, but not the distance to the landmarks. He developed an algorithm for performing localization from this data in $O(n^3)$ time, where n is the number of landmarks. Extensions of this method yielded an $O(n^2 \log n)$ algorithm for a robot with a compass and an $O(n^2)$ algorithm for the case where the landmarks are distinguishable. Betke and Gurvits [4] further consider the case where the landmarks are distinguishable. By representing the landmark positions as complex numbers, they obtained a linear time algorithm with a least-squares error criterion.

Another localization technique is to use a search tree [13, 17] to perform matching between features or landmarks detected by a sensor and the known map. Drumheller [11] used this technique to perform localization using walls detected by sonar. He also uses a *sonar barrier test* to check for inconsistencies based on the constraints of sonar data. Simsarian *et al.* [36] use a variation of this technique where the map is decomposed into *view-invariant regions*, which are used to guide the tree search and reduce the cost of feature matching. Cox [8] also performed matching between line segments in the plane, although detection was performed with a laser range-finder. Cox assumed that the robot would have rough knowledge of its location and thus used an iterative least-squares fitting procedure to improve the position estimation.

Elfes [12] used an occupancy grid representation of the environment. Each cell in the grid

was given a score between -1 and 1, where -1 represents unoccupied, 1 represents occupied, and values in between represent varying levels of certainty. Localization was performed by locating the position between a local and a global occupancy grid that maximized the product of the values at the corresponding cells in the grids.

Atiya and Hager [1] address a problem where the landmarks are two-dimensional points on a plane (e.g. vertical edges sensed with stereo vision). Correspondences are determined by matching triples of sensed landmarks to triples of map landmarks, since such triples yield lengths and angles that are invariant to the robot position. Uncertainties in the localization estimate are computed by intersecting the uncertainty regions of the landmarks, which are approximated by rectilinear cells.

A technique that has been used for rough localization in a large environment is to examine the features present on the horizon and to use some strategy to match them to the known elevation map of the terrain. This technique can provide coarse localization in very large environments when there is no knowledge of the robot's position, but it does not provide fine localization for positioning. Talluri and Aggarwal [38] use the shape of the horizon line to search for the position of a robot in a digital elevation map. They first perform pruning using geometrical constraints to eliminate many positions in a discretized space of possible robot positions. For positions that pass the first stage, a refinement step is used that performs curve matching between the visible horizon and the estimated horizon line computed from the elevation map. The best match is taken to be the most likely robot position. Stein and Medioni [37] approximate the horizon line by a polygonal chain and index a table storing subsections of the horizon as it would be seen from each position in a discretized pose space on the map. A verification step for the indexed matches uses geometric constraints to select the best match. Thompson *et al.* [40] extract and match features on the horizon and other visible hills and ridges. Matches between configurations of features are then searched for in a map that has been pre-processed. The hypothesized locations are then refined and evaluated. Cozman and Krotkov [9] also detect mountain peaks on the horizon. However, they perform the search in the discretized space of positions using table look-up in order to maximize the posterior probability of finding the correct position.

A final technique that we will mention is to perform localization by matching a three-dimensional map of the terrain near the robot to a previously generated map. This is the approach that Kweon and Kanade [22] take in order to generate a terrain map by fusing multiple local maps. They first generate a terrain map from stereo vision using the "locus method". They then perform matching between the maps in a two-stage procedure. First, an estimate for the relative position is generated by extracting and matching map features (high curvature points). The estimate is then refined using a iterative optimization procedure. Zhang [42] also describes a technique that can be used for matching 3-D terrain maps. His technique uses an initial estimate of the relative position between two sets of points to iteratively improve the estimated position. At each iteration, the technique determines the closest match for each point and updates the estimated position based on a least-squares metric, with some modifications to increase robustness.

3 Map similarity measure

As we have previously noted, we perform localization by matching a map generated at the current robot position (the *local map*) to a previously generated map of the environment (the *global map*), which may be generated by combining previous local maps. The optimal position of the robot with respect to the global map is located using a maximum-likelihood similarity measure for comparing images and maps [30]. This similarity measure (which is described below in more detail) yields a score for each possible position of the local map with respect to the global map, given a prior probability distribution of robot positions and a probability distribution function (PDF) for the distance from the features in the local map to the closest feature in the global map. This PDF models the sensor uncertainty and the possibility of missing a feature. When an appropriate PDF is used to model the sensor uncertainty, this measure is robust to outliers, noise, and occluded locations. In addition, it can be applied to maps consisting of sparse landmarks or to a dense occupancy map representation.

In order to formulate the map matching problem in terms of maximum likelihood estimation, we must have some set of measurements that are a function of the robot position. We use the distances from the visible features at the current robot location to the closest features in the global representation of the environment. The method by which these distances are computed is problem dependent. We have simply used the Euclidean distance for both landmarks and occupancy maps, but more complex distance functions, such as the Mahalanobis distance can be used given the requisite covariance information. Since we search for the best relative position between these maps, these distances are variables.

Let us say that our local map L consists of n features $\{l_1, ..., l_n\}$ and that our global map G consists of m features $\{g_1, ..., g_m\}$. These features may represent individual landmarks or the may represent cells in an occupancy grid. The distance between a feature l_i in the local map and a feature g_i in the global map when the local map is at position X with respect to the global map is denoted $d_{ij}^X = \text{dist}(X(l_i), g_j)$, where X can be thought of as a function that transforms features in the local map into their corresponding position in the global map. The distance from a feature in the local map to the closest feature in the global map (at some relative position X between the maps) is D_i^X.

$$D_i^X = \min_{1 \leq j \leq m} d_{ij}^X \tag{1}$$

While these distances are not completely independent, we have found that modeling them as such yields very good results. Recent work on determining the probability of a false positive for matching sparse features (such as landmarks) [14, 16] and for matching dense features (such as edge maps and occupancy grids) [15, 31] has also achieved accurate results when treating the features independently. We thus formulate the likelihood function for the robot position, X, as the product of prior probability of the position with the probability distributions of these distances:

$$P(X) = p(X) \prod_{i=1}^{n} p(D_i^X) \tag{2}$$

For convenience, we work in the $\ln P(X)$ domain, since this does not change the relative ordering of the positions:

$$\ln P(X) = \ln p(X) + \sum_{i=1}^{n} \ln p(D_i^X) \tag{3}$$

The position yielding the maximum likelihood is taken to be the position of the robot. The prior probability distribution of robot positions and the probability distribution function (PDF) that is used for each feature, $p(D_i^X)$, together determine the matching measure that is used between the maps. If nothing is known about the prior distribution of model positions, then it can be modeled by a constant and removed from the measure. On the other hand, if we are tracking the robot position over time (e.g. with an extended Kalman filter), we will have some known prior (a normal distribution in the case of the EKF) and this will affect the computed position of the robot. We must also estimate the PDF of the feature distances. A two-valued PDF with a constant prior yields a measure equivalent to the Hausdorff fraction (which is a commonly used measure for image matching [30]):

$$\ln p(D_i^X) = \begin{cases} k_1 + k_2, & \text{if } D_i^X \leq \delta \\ k_1, & \text{otherwise} \end{cases} \tag{4}$$

This PDF is simple and yields good results, but does not accurately model the distribution of distances unless they are uniformly distributed both inside and outside of the specified error boundary. The actual values of k_1 and k_2 do not affect the ranking of the positions (as long as $k_2 > 0$). In practice, we have used $k_1 = 0$ and $k_2 = 1$.

Better results can be achieved through the use of a more complex PDF that accurately models the sensor uncertainty [30]. The feature localization errors can often be accurately modeled by a normal distribution. However, this does not allow for outliers in the local feature map, which have no corresponding features in the global map. The use of a normal distribution with a constant additive term yields an accurate model for cases with outliers [28]:

$$p(D_i^X) = \max_{g \in G} \; k_1 + \frac{k_2}{\sigma\sqrt{2\pi}} e^{-(D_i^X)^2/2\sigma^2} \tag{5}$$

In this PDF, σ is the standard deviation of the feature uncertainty and k_1 and k_2 are constants that vary with the frequency of outliers, the density of the maps, and the probability of missing a feature. The robot localization is insensitive to the settings of these constants, but a complete analysis of the values these constants should take can be found in [28].

4 Finding the most likely position

Now that the similarity measure between the maps has been defined, we must discuss how the position the optimizes the similarity measure is determined. A simple hill-climbing technique could be used, but such a method would require good initial estimate of the position of the robot, which is not always available, particularly if we exercise the localization techniques infrequently. We describe a method to search a bounded pose space using a variation of branch-and-bound search that guarantees that we locate the best position in a discretized version of the search space. For some map representations and appropriate discretizations, this will be the optimal position over the entire search space. A subsequent subpixel localization step can also be performed to gain accuracy. Following the general discussion of the search strategy, we discuss the application of this search strategy to maps consisting of landmarks and occupancy grids in more detail.

4.1 Search strategy

We locate the most likely robot position by adapting a multi-resolution search strategy that has been applied to image matching using the Hausdorff distance [21, 31, 34]. We first test the nominal position of the robot given by dead-reckoning (or any other position, if no initial estimate is available) so that we have an initial position and likelihood to compare against. Next, the pose space is divided into rectilinear cells. Each cell is tested using a conservative test to determine whether it could contain a position that is better than the best position found so far. Cells that cannot be pruned are divided into smaller cells, which are examined recursively. When cells of a certain (small) size are reached, the cells are tested explicitly. For example, when we compare occupancy grids under translation, there is a natural discretization of the pose space such that neighboring positions move the maps by one grid cell with respect to each other. For this case, we stop dividing the cells when they contain a single position in this discretization and we test this position explicitly. For more complex examples, we may set some threshold on the minimum cell size and test the center of the cell when the cell size is below the threshold. Subpixel localization estimates will be useful in increasing the accuracy of the localization in this case (see Section 5).

The key to this strategy is a quick method to test the cells. A cell is allowed to pass the test if it does not contain a good pose, but it should never prune a cell that might contain a good pose, since this could result in the best position being missed. To determine whether a particular cell C could contain a pose that is superior to the best one found so far, we examine the pose c at the center of the cell. In order to place a bound on the best position within the cell, we compute the maximum distance between the location to which a feature in the local map is transformed into the global map by c and by any other pose in the cell. Denote this distance Δ_C. If we treat robot poses as functions that transform positions in the local map into positions in the global map then Δ_C can be written:

$$\Delta_C = \max_{p \in C} \max_{l \in L} ||p(l) - c(l)|| . \tag{6}$$

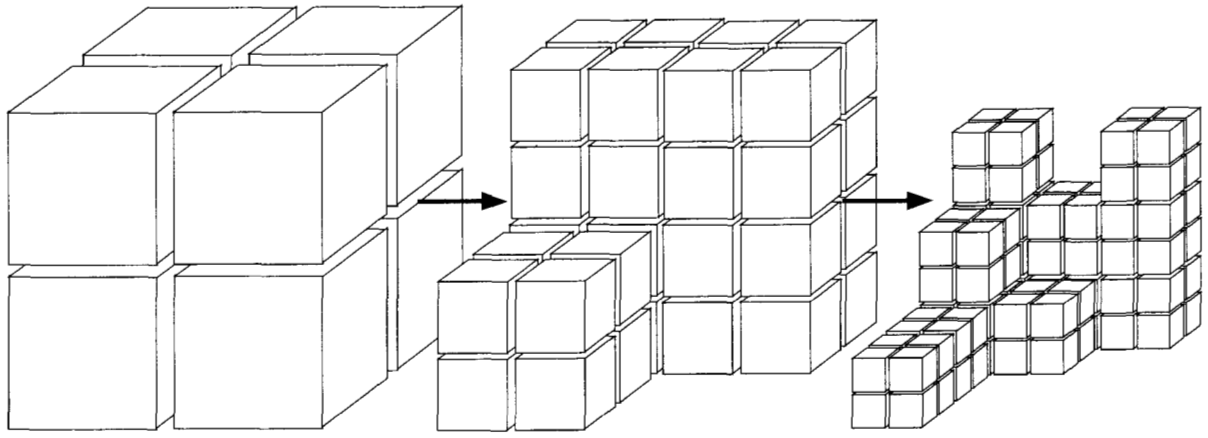


Figure 2: A search strategy is used that recursively divides and prunes cells of the search space.

For the space of translations, Δ_C is simply the distance from the center of the cell to any corner of the cell, since the difference in the translated location of any feature in the local map for any two translations is simply the difference between the translations. When rotations are considered, Δ_C is also a function of the local map. In this case, Δ_C can be computed as a function of maximum orientation change between the center of the cell and the corners of the cell. We concentrate on translations of the robot, since the robot orientation can often be determined through other sensors. Further discussion of techniques to handle rotations in such a branch-and-bound search strategy can be found elsewhere [31, 34].

To place a bound on the quality of any position within the cell, we bound each of the distances that can be achieved by features in the local map over the cell and then propagate them into the likelihood function.

$$D_i^C = \max(D_i^c - \Delta_C, 0) \tag{7}$$

$$P_i^C = \ln p(D_i^C) \tag{8}$$

D_i^c is the distance from the ith feature in the local map at the position given by c, the center of the cell, to the closest feature in the global map. D_i^C is a bound on the distance that can be achieved for the ith feature at any position in the cell C. P_i^C is the maximum score that the ith feature of the local map can contribute to the likelihood for any position in the cell[1].

A bound on the best overall likelihood that can be found at a position in the cell is given by:

$$\max_{X \in C} P(X) \leq \sum_{i=1}^{n} P_i^C \tag{9}$$

[1]This assumes that the PDF is monotonically non-increasing, which is true for any reasonable PDF, since we desire closer matches to yield higher scores.

If this likelihood does not surpass the best that we have found so far, then we can prune the entire cell from the search. Otherwise the cell is divided into two cells of the same size by slicing it along the longest axis and the process is repeated recursively until all of the cells have been exhausted.

4.2 Occupancy grids

The search strategy described above is well suited to matching maps that are represented by occupancy grids, since these are inherently discretized. The space of translations of the robot is discretized at the same resolution as the maps and this yields a natural resolution of the search space at which to end the recursive division of the cells.

The simplest case is a grid representation where each cell represents either occupied or unoccupied space. This allows for a fast implementation of the search, since each D_i^c can be computed efficiently over the entire global map by computing the distance transform of the map. The distance transform measures the distance from each cell in a discretized map to the closest occupied cell [33], and can be computed efficiently using an algorithm that is linear in the size of the map [6, 7].

In order to implement this procedure efficiently, we first compute the distance transform of the global occupancy map. We can then compute a relative index into the distance transform for each occupied cell in the local map. The pose-space cell hierarchy is searched using a depth-first search strategy. For each cell that is examined, we loop through the precomputed indexes into the distance transform (which must be offset by position of the center of the cell). For each index, we get a distance in the global map. We then use Equations (7), (8), and (9) to determine whether the cell can be pruned.

For probabilistic occupancy grids, the strategy is slightly more complex to implement. A typical representation will use scores in the range $[-1, 1]$ for each cell of the grid. In this case, -1 represents the case where the cell is known to be unoccupied, 1 represents the case where the cell is known to be occupied, and the intermediate states represent various levels of uncertainty. See, for example, [12]. We may, of course, map such a grid into a binary representation, but this results in lost information. An alternative solution is to use the product of the occupancy grid scores between the local map and the global map at each position as the likelihood score for the position [12]. In this case, we can examine a dilated version of the global map in order to decide whether to prune a cell in the pose space. Let Δ_C be the radius of the cell as defined above and $G(X)$ be the occupancy grid representing the global map. We can write the dilation as follows:

$$G_{\Delta_C}(X) = \max_{||X'-X|| \leq \Delta_C} G(X') \tag{10}$$

Each position X in the new grid thus contains the maximum score over a circle centered at X with radius equal to Δ_C.

Now we sum over the locations in the local map for which we have some knowledge to

obtain the bound on the likelihood for any position in the cell:

$$\max_{X \in C} P(X) \leq \sum_{i=1}^{n} L(l_i) G_{\Delta_C}(c(l_i)) \tag{11}$$

If the sum is smaller than the best found so far, then we can prune the cell.

In order to implement the search efficiently we use a breadth-first search of the cell hierarchy in the pose space, making sure that all of the cells at each level of the search have the same dimensions. In this case, Δ_C is the same for each of the cells at any level of the search tree and we can compute the dilation described above once for each such level.

4.3 Landmarks

These techniques can also be easily applied to matching maps consisting of geometric landmarks. For example, in indoor environments we may be able to detect and locate vertical edges, or we may use the peaks of rocks or other landmarks in outdoor terrain. In this case, we can use efficient nearest-neighbor searching techniques to compute each D_i^c exactly. For example, we may use the methods of Lipton and Tarjan [24] or Bentley [3] to locate the nearest landmark, if the landmarks are represented by points, and the distance can then be computed directly.

These techniques can be made even more efficient at the cost of a small amount of accuracy by discretizing the landmark positions. In this case, the distances can be computed using the distance transform of the map, as described above. We can then use subpixel localization techniques to improve the accuracy over the position yielded by the discretized search space (see below). Our experiments have indicated that the amount of accuracy lost is quite small when using this technique.

Once the method of computing each D_i^c is determined, the remainder of the search strategy is same as described above.

5 Subpixel localization and uncertainty estimation

Using this probabilistic formulation of the localization problem, we can estimate the uncertainty in the localization, in terms of both the variance of the estimated positions and the probability that a qualitative failure occurred. In addition, we can perform subpixel localization in the discretized pose space by fitting a surface to the peak that occurs at the most likely robot position.

5.1 Subpixel localization

Let us take as an assumption that the likelihood function approximates a normal distribution in the neighborhood around the peak location. Fitting such a normal distribution to the computed likelihoods yields both an estimated variance in the localization estimate and a

subpixel estimate of the peak location. While the approximation of the likelihood function as a normal distribution may not be accurate in general, it does yield a good fit to the local neighborhood around the peak and our experimental results indicate that very accurate results can be achieved under this assumption.

Now, since we actually perform our computations in the domain of the natural logarithm of the likelihood function, we must fit these values with a polynomial of order 2. If we assume independence in x and y, then we have:

$$\ln P(x,y) = \ln \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}} = -\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y} \tag{12}$$

In order to estimate the parameters that we are interested in (x_0, y_0, σ_x, and σ_y), we project this polynomial onto the lines $x = x_0$ and $y = y_0$, yielding:

$$P(x,y_0) = -\frac{(x-x_0)^2}{2\sigma_x^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y} \tag{13}$$

$$P(x_0,y) = -\frac{(y-y_0)^2}{2\sigma_y^2} + \ln \frac{1}{2\pi\sigma_x\sigma_y} \tag{14}$$

We now fit these equations to the x and y cross-sections of the likelihood function at the location of the peak. If the peak in the discretized search space occurs at position (x_p, y_p), we fit $P(x, y_0)$ to the values at the surrounding 5 positions along $y = y_p$:

$$p_{-2} = P(x_p - 2, y_p) \tag{15}$$

$$p_{-1} = P(x_p - 1, y_p) \tag{16}$$

$$p_0 = P(x_p, y_p) \tag{17}$$

$$p_1 = P(x_p + 1, y_p) \tag{18}$$

$$p_2 = P(x_p + 2, y_p) \tag{19}$$

The least-squares fit to a parabola ($y = ax^2 + bx + c$) with $x = \{-2, -1, 0, 1, 2\}$ yields:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{1}{7} & -\frac{1}{14} & -\frac{1}{7} & -\frac{1}{14} & \frac{1}{7} \\ -\frac{1}{5} & -\frac{1}{10} & 0 & \frac{1}{10} & \frac{1}{5} \\ -\frac{3}{35} & \frac{12}{35} & \frac{17}{35} & \frac{12}{35} & -\frac{3}{35} \end{bmatrix} \begin{bmatrix} p_{-2} \\ p_{-1} \\ p_0 \\ p_1 \\ p_2 \end{bmatrix} \tag{20}$$

We can now solve for x_0 and σ_x using:

$$x_0 = x_p - \frac{b}{a} \tag{21}$$

$$\sigma_x = \frac{1}{\sqrt{-2a}} \tag{22}$$

The derivation for y_0 and σ_y is the same, except that we project onto the line $x = x_p$. The values of x_0 and y_0 yield a subpixel localization result, since this is the estimated location of the peak in the likelihood function. In addition, σ_x and σ_y now yield direct estimates for the uncertainty in the localization result.

5.2 Probability of failure

In addition to estimating the uncertainty in the localization estimate, we can use the likelihood scores to estimate the probability of a failure to detect the correct position of the robot. This is particularly useful when the terrain yields few landmarks or other references for localization and thus many positions appear similar to the robot.

For a discretized search space, we can estimate this probability of failure by summing the likelihood scores for the peak selected as the most likely robot position and comparing to the sum of the likelihood scores that are not part of this peak. In practice, we can usually estimate the sum under the peak by examining a small number of values around the peak, since they fall off very quickly (recall that the computed values are the logarithm of the likelihood function). The remainder of the values can also be estimated efficiently. Whenever a cell in the search space is considered, we compute not only a bound on the maximum score that can be achieved, but also an estimate on the average score that is achieved by determining the score for the center of the cell. If the cell is pruned, then the sum is incremented by the estimated score multiplied by the size of the cell. In practice, this yields a very good estimate, since cells with large scores cannot be pruned until they become small. We thus get good estimates when the score is significant and when the estimate is not as good, it is because the score is small and does not significantly affect the overall sum.

Let S_p be the sum obtained for the largest peak in the pose space and S_t be the overall sum for the pose space. We can estimate the probability of correctness for the largest peak as:

$$P_c = \frac{S_p}{S_t} \tag{23}$$

6 Experiments

We have applied these techniques in a number of experiments using both synthetic data, where precise ground-truth was available for comparison, and real range data from stereo vision, including experimental localization results for the Sojourner rover on Mars.

6.1 Synthetic landmarks

We first applied these techniques to localization using landmarks in synthetic experiments. In these experiments, we randomly generated a synthetic environment containing 160 landmarks on a 256×256 unit square. Let us say that each unit is 10 cm (though the entire problem scales to an arbitrary size). In each trial, seven of the ten landmarks closest to some random

robot location were considered to be observed by the robot (with Gaussian error in both x and y with standard deviation $\sigma = 1$ unit) along with 3 spurious landmarks not included in the map. Localization was then performed using these 10 observed landmarks with no knowledge of the position of the robot in this environment. Over 100000 trials, the robot was correctly localized in 99.8% of the cases, with an average error in the correct trials of 0.356 units in each dimension. The average estimated standard deviation in the localization using the techniques from the previous section was 0.427 units.

Figure 3(a) shows the distribution of actual errors observed versus the distribution that we expect from the average standard deviation estimated in the trials. The close similarity of the plots indicates that the estimated standard deviation is a very good estimate of the actual value. It appears that this estimate is slightly smaller than the true value since the frequency of the observed errors is slightly above the curve at the tails and lower at the peak. However, the overall similarity is very good. The similarity between these plots also validates the approximation of the likelihood function as a normal distribution in the neighborhood of the peak. Figure 3(b) shows the distribution of the estimated standard deviations in this experiment. It can be observed that the estimate is very consistent between trials, since the plot is very strongly peaked near the location of the average estimate. The right tail of the plot is longer than the left tail, indicating that when errors occur they are more likely to overestimate the standard deviation of the error. Taken together, these plots indicate that the standard deviation estimates are very likely to be accurate for each individual trial.

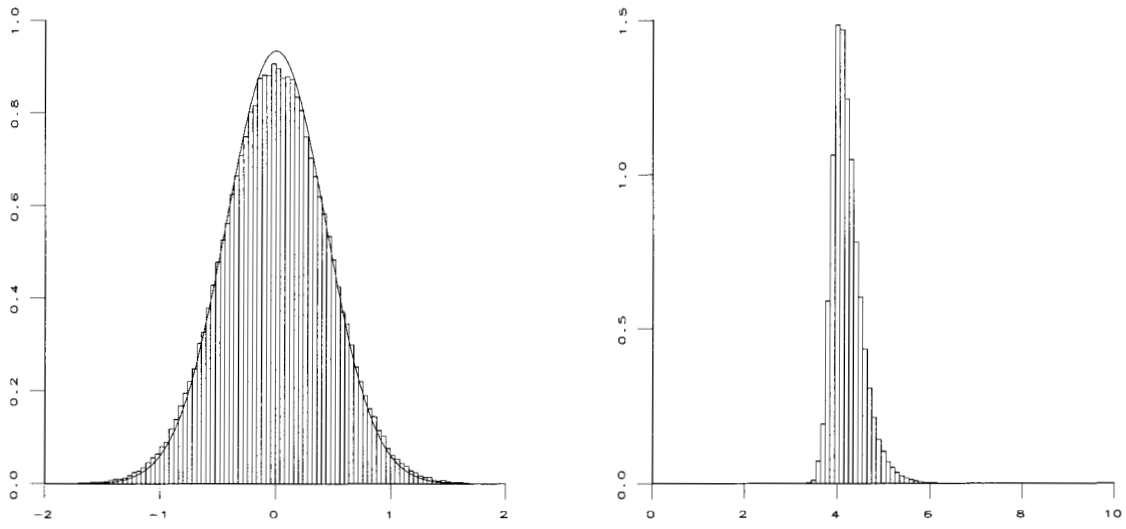
We also tested the probability of correctness measure in these trials. The average probability of correctness computed for the trials that resulted in the correct localization was .993, while the average probability of correctness for the failures was .643. The probability of correctness measure thus yields information that can be used to evaluate whether the localization result is reliable.

6.2 Localization using stereo range data

In practice, we have performed matching between three-dimensional occupancy maps to perform localization for planetary rovers. For these occupancy maps, we have considered every cell to be either occupied or unoccupied (with no in-between states). While several methods could be used for generating such a representation, we have used stereo vision on the Rocky 7 rover [18] to compute range maps using the techniques that have been previously described by Matthies [25, 26].

Once a range map has been computed from the stereo imagery, we convert it into a voxel-based map representation. We first rotate the data such that it has the same relative orientation as the map we are comparing it to. Here we operate under the assumption that the orientation of the robot is known through sensors other than vision (for example, both Sojourner and Rocky 7 have rate gyros and accelerometers and Rocky 7 also uses a sun sensor for orientation determination [41]). The localization techniques can also be generalized to determine the robot's orientation.

The next step is to bin the range points in a three-dimensional occupancy map of the

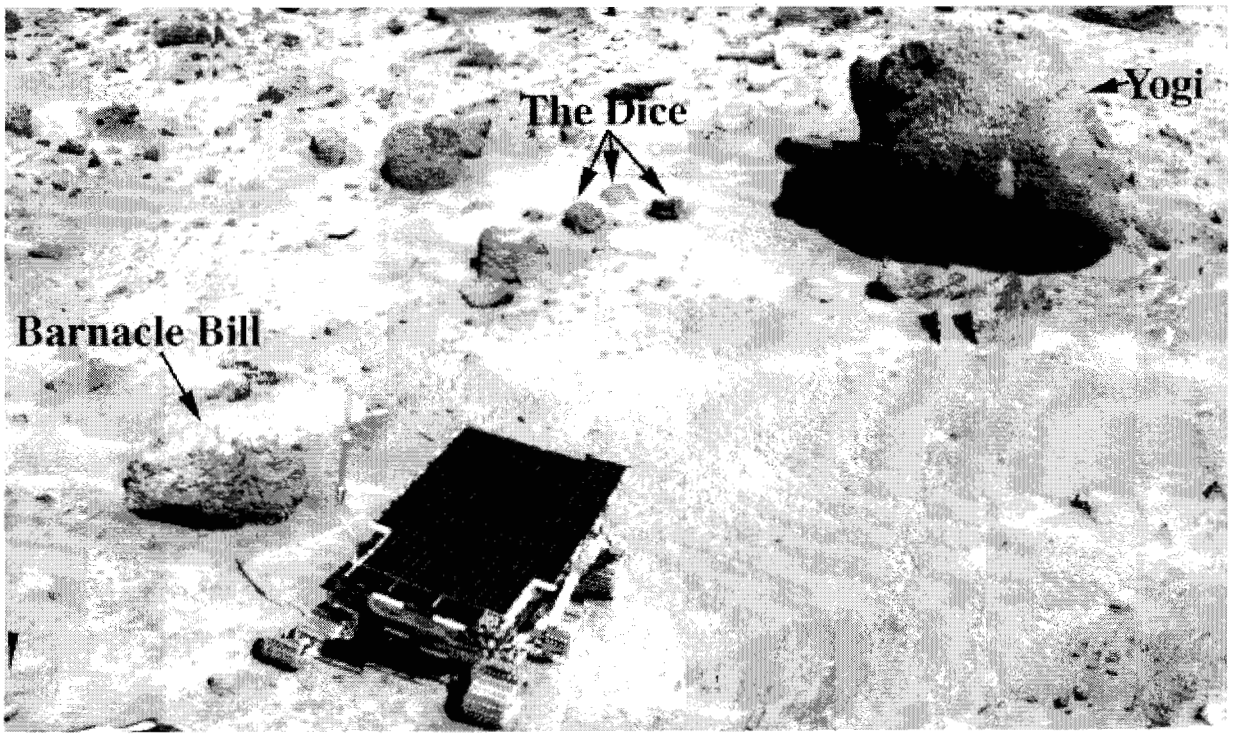


(a) (b)

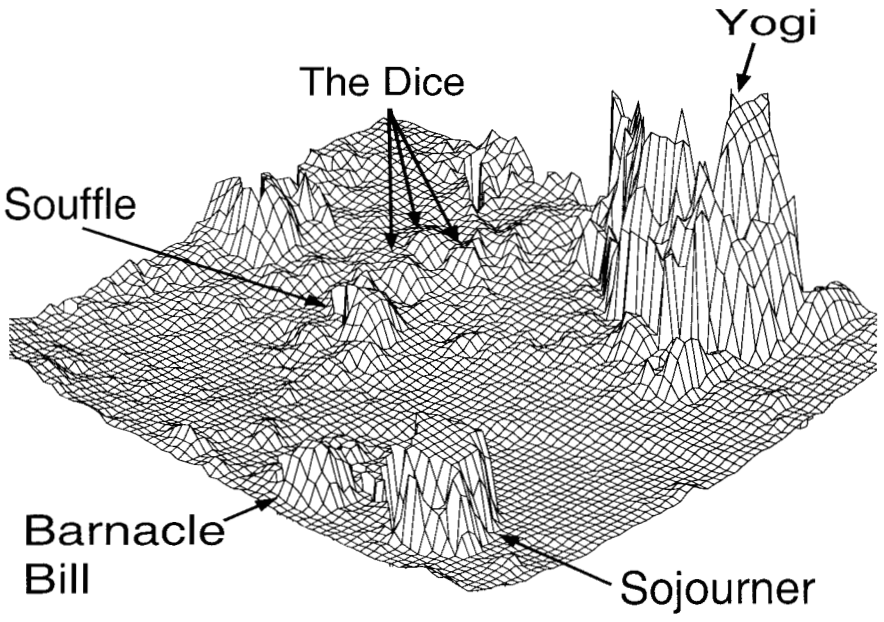
Figure 3: Distribution of errors and estimated standard deviations in synthetic landmark localization experiment. (a) Comparison of estimated distribution of localization errors (solid line) to observed distribution of localization errors (bar graph). (b) Distribution of estimated standard deviations in the localization estimate.

surroundings at some specified scale. We eliminate the need to search over the possible translations of the robot in the z-direction by subtracting a local average of the terrain height from each cell (i.e. a high-pass filter). This step is not strictly necessary, and it reduces our ability to determine height changes in the position of the robot, but it also reduces the computation time that is required to perform localization. A subsequent step can be performed to determine the robot elevation, if desired. Each cell in the occupancy map that contains a range pixel is said to be *occupied*, and the others are said to be *unoccupied*. Figure 4 gives an example of a terrain map that was generated using imagery from the Mars Pathfinder mission.

We have tested these techniques using both terrestrial data and data from the Mars Pathfinder mission. The results indicate that self-localization can performed with these techniques approximately as well as a human operator, without requiring a downlink cycle. In addition, these techniques require only a few seconds to perform localization. Experiments indicate that localization can be performed on a SPARCstation 20 in under 5 seconds with maps discretized at 2 cm resolution. Similar experiments performed on-board Rocky 7 (Motorola 68060 CPU) require approximately 20 seconds once the stereo range image has been computed.



(a)



(b)

Figure 4: Terrain map generated from Pathfinder imagery. (a) Annotated composite image of Sojourner and rocks on Mars. (b) Terrain map generated from stereo imagery.

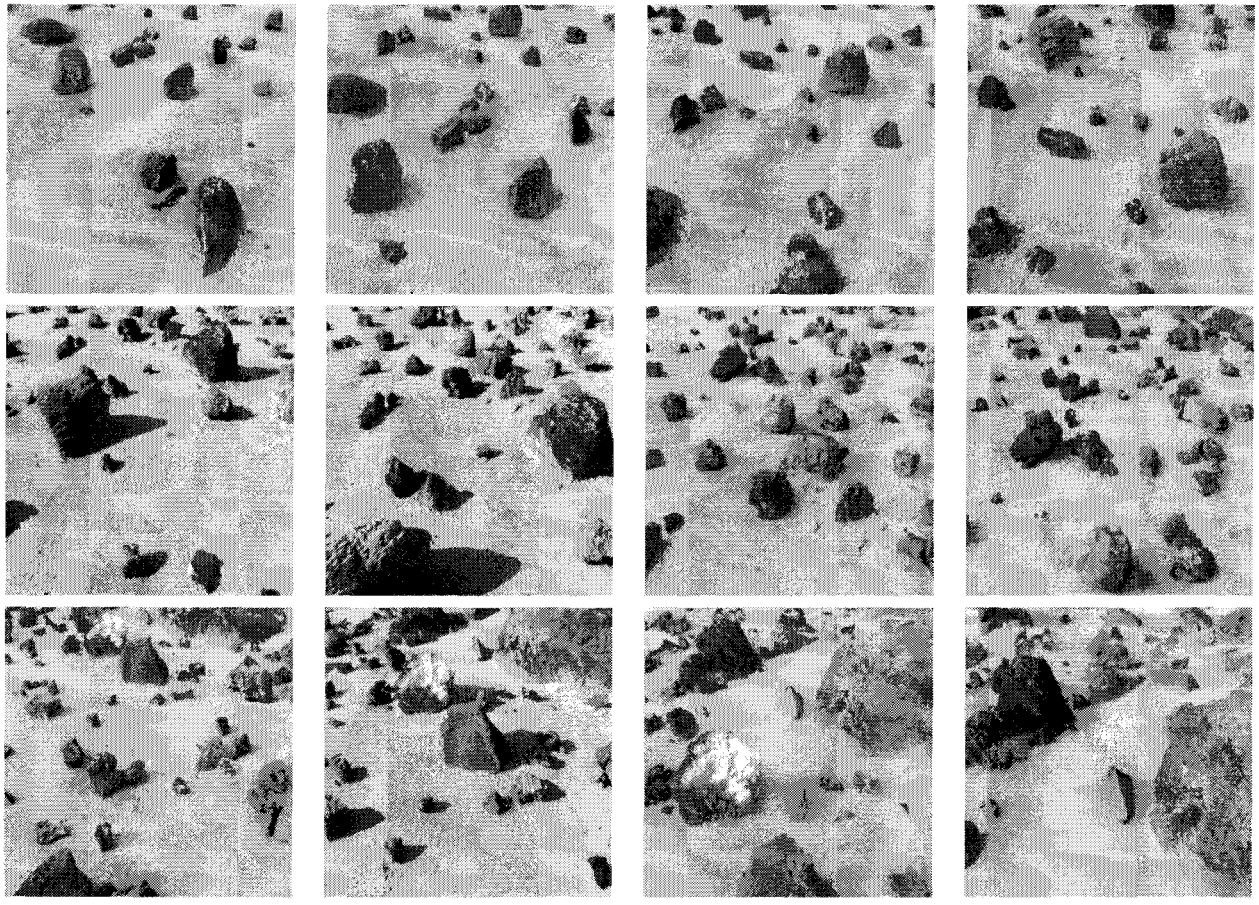


Figure 5: A sequence of images used for testing the localization techniques.

6.3 Tripod-mounted cameras

We initially tested these techniques with images taken in the JPL Mars Yard[2] using cameras mounted on a tripod at approximately the Rocky 7 mast height. Figure 5 shows a set of images that was used in testing the localization techniques. The set consists of 12 stereo pairs acquired at one meter intervals along a straight line with approximately the same heading.

In these tests, we determined the estimated position changes by finding the relative position between each pair of consecutive images. These relative positions were determined by matching occupancy maps created as described above. The localization techniques yielded a qualitatively correct position between each pair of consecutive images. The average absolute error in the position estimates was 0.0342 meters in the downrange direction and 0.0367 meters in the crossrange direction from the position measured by hand. Much of this error is attributable to human error in determining the ground truth for the data.

[2]See `http://robotics.jpl.nasa.gov/tasks/scirover/marsyard`

Additional tests were performed on imagery where the camera system was panned 25 degrees left and right. In these tests, occupancy maps from the panned images were matched to occupancy maps for the unpanned images. All 24 trials yielded the correct qualitative result. The average absolute error was 0.0138 meters in the downrange direction and 0.0225 meters in the crossrange direction.

In these tests, the average number of positions examined was 18.45% of the total number of positions in the discretized search space. A speedup of greater than 5 was thus achieved through the use of the efficient search techniques.

6.4 Pathfinder

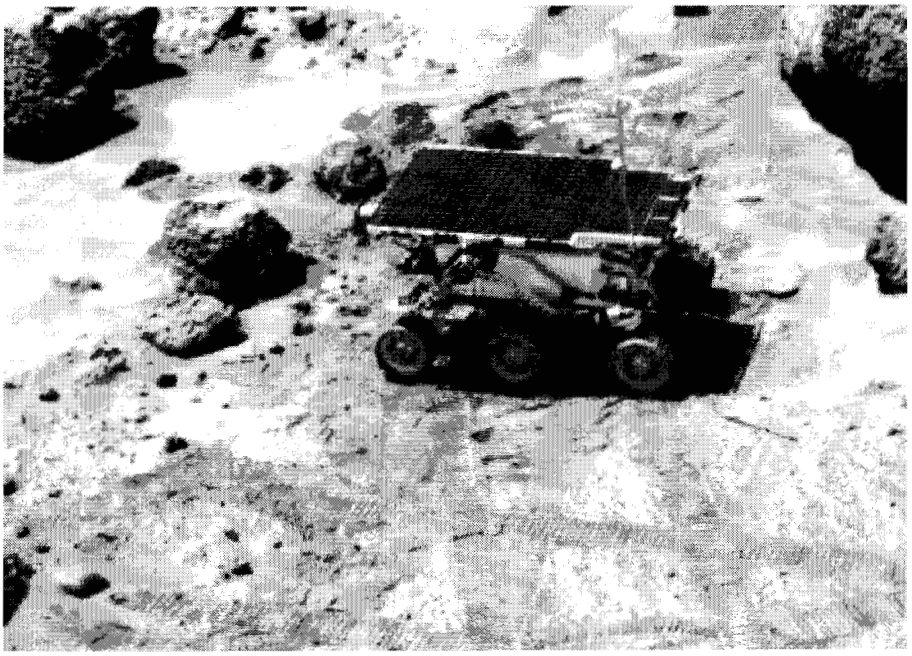
To validate these techniques for use on a Mars rover, we have tested them using data from the Mars Pathfinder mission. A map of the terrain surrounding the Pathfinder lander was first generated using stereo imagery. For each position of Sojourner at which we tested the localization techniques, we generated an occupancy map of the terrain using range data from Sojourner's stereo cameras. This local map was then compared to the global map from the lander.

Unfortunately, this test has only been possible at a few locations due to the limited amount of data returned to Earth, the lack of interesting terrain in some of the imagery we do have, and the lack of a comparison value for most positions (except those where Sojourner was imaged by the lander cameras). In practice, these techniques could be exercised much more frequently since they would not require downlinking image data to Earth and the comparison value is only necessary for testing. We envision a scenario where the data from the rover's navigation cameras, which would be operating frequently in order to perform obstacle detection, would be used to perform localization whenever sufficient terrain was evident in the imagery. In addition, the imagery from mast cameras could be used for localization when the positional uncertainty grows beyond the desired level and the imagery from the navigation cameras is unsuitable.

As an example of the data, Figure 6 shows the position of Sojourner as seen from the lander and the view from Sojourner at the end of sol 21[3] of the Mars Pathfinder mission. Note that the stereo data obtained from Sojourner is not as good as we hope to achieve in future missions. Accurate stereo data is achieved only for the central portion of the Sojourner imagery due to inaccurate calibration of the fish-eye lenses. The field-of-view that we have to work with is thus relatively small. However, we have achieved good localization results with this data.

Table 1 shows the results of localization using the techniques described in this paper versus the localization that was obtained by human operator through overlaying a rover model on the stereo data obtained from imaging the rover from the lander. For sol 42, we have two localization results, one prior to and one after a turn by the rover. The operator localization was performed after the turn.

[3] A *sol* is a Martian day



(a)

(b)

Figure 6: Sojourner on sol 21 (near "Souffle"). (a) Composite image from the lander. (b) Image from Sojourner.

Sol	Operator		Localization	
	x	y	x	y
4	3.28	-2.69	3.01	-2.64
10	4.34	-3.24	4.24	-3.27
21	3.32	-2.60	3.37	-2.65
27	-5.42	2.85	-4.98	2.75
42a	-3.00	-1.86	-3.02	-1.87
42b	-3.00	-1.86	-3.00	-1.87
72	-8.93	-1.57	-8.99	-1.35

Table 1: Comparison of rover positions determined by a human operator overlaying a rover model on stereo data of the rover and by the localization techniques described in this paper.

The results show very close agreement between our techniques and the operator localization for four of the sols. For sols 4, 27, and 72, there is some disagreement. Possible sources of error include inaccurate calibration of either the rover or lander cameras and operator error in performing localization. Manual examination of the maps indicates that the localization techniques determine the qualitatively correct position in these cases. While no ground truth exists, the similarity of the positions estimated by these techniques and by the human operator indicate that these techniques can perform localization approximately as well as a human operator.

7 Summary

We have described a method for performing self-localization for mobile robots through maximum-likelihood matching of maps. The map of visible features at the robot's current position is compared to a global map that has been previously generated (possibly by combining the maps from the robot's previous positions). The best relative position between the maps is detected using a global branch-and-bound search technique that does not require an initial estimate of the robot position. The search is performed relative to a maximum-likelihood map similarity measure that selects the robot position at which the maps best agree.

This probabilistic formulation of the map matching problem allows the uncertainty in the localization of individual map features to be treated accurately in the matching process. In addition, performing a polynomial fit to the log-likelihood function allows both subpixel localization to be performed and uncertainty estimates to be computed, which can be propagated in a position tracking mechanism such as the extended Kalman filter.

Our goal in the design of these techniques is to provide greater autonomy for Mars rovers. Through the use of these techniques we can perform self-localization on Mars within the confines of a science site where panoramic stereo imagery has been taken from the lander or from

the rover mast cameras. These techniques can also be used to improve position estimation on long traverses by periodically stopping to perform localization versus the previous position and to image the terrain ahead of the rover. The application of these techniques to data from the Mars Pathfinder mission indicates that we can perform autonomous localization with approximately the same accuracy as a human operator without requiring communication with Earth.

An area that bears further study is the development of a localizability measure for terrain maps in order to plan effective localization steps. In the future, we also plan to integrate these techniques into an integrated navigation methodology, in which a Kalman filter is used to synthesize a robot position estimate from a variety of sensors and the robot's path planner interacts with the Kalman filter and the localization techniques to plan when and where localization should be performed.

Acknowledgements

This work is an element of the Long Range Science Rover project, which is developing technology for future Mars missions. This project is funded as part of the NASA Space Telerobotics Program, through the JPL Robotics and Mars Exploration Technology Office. The author would like to thank all of the people who have been involved with the Mars Pathfinder mission and the Long Range Science Rover project. In particular, thanks are due to Larry Matthies for generating the range maps from the Pathfinder IMP cameras that were used in some of the experiments reported here and to Sharon Laubach for collecting the data in the Mars yard used in other experiments reported here. Finally, the author thanks the people who have read and commented on various versions of this work, including Brian Wilcox, Samad Hayati, Mark Maimone, Andrew Johnson, and the anonymous reviewers.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government, or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] S. Atiya and G. D. Hager. Real-time vision-based robot localization. *IEEE Transactions on Robotics and Automation*, 9(6):785–800, December 1993.

[2] N. Ayache and O. D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, December 1989.

[3] J. L. Bentley, B. W. Weide, and A. C. Yao. Optimal expected time algorithms for closest point problems. *ACM Transactions on Mathematical Software*, 6:563–580, 1980.

[4] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, April 1997.

[5] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 14(4):231–249, 1997.

[6] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[7] H. Breu, J. Gil, D. Kirk, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, May 1995.

[8] I. J. Cox. Blanche - An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, April 1991.

[9] F. Cozman and E. Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proceedings of the IEEE Conference on Robotics and Automation*, volume 3, pages 2452–2457, 1997.

[10] A. Curran and K. J. Kyriakopoulos. Sensor-based self-localization for wheeled mobile robots. *Journal of Robotic Systems*, 12(3):163–176, 1995.

[11] M. Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332, March 1987.

[12] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

[13] P. C. Gaston and T. Lozano-Pérez. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):257–265, May 1984.

[14] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255–274, March 1990.

[15] W. E. L. Grimson and D. P. Huttenlocher. Analyzing the probability of a false alarm for the Hausdorff distance under translation. In *Proceedings of the Workshop on Performance versus Methodology in Computer Vision*, pages 199–205, 1994.

[16] W. E. L. Grimson, D. P. Huttenlocher, and D. W. Jacobs. A study of affine matching with bounded sensor error. *International Journal of Computer Vision*, 13(1):7–32, 1994.

[17] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[18] S. Hayati et al. The Rocky 7 rover: A Mars sciencecraft prototype. In *Proceedings of the IEEE Conference on Robotics and Automation*, volume 3, pages 2458–2464, 1997.

[19] J. Horn and G. Schmidt. Continuous localization of a mobile robot based on 3d-laser-range-data, predicted sensor images, and dead-reckoning. *Robotics and Autonomous Systems*, 14:99–118, 1995.

[20] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[21] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 705–706, 1993.

[22] I. S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278–292, February 1992.

[23] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[24] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

[25] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8(1):71–91, July 1992.

[26] L. Matthies, A. Kelly, T. Litwin, and G. Tharp. Obstacle detection for unmanned ground vehicles: A progress report. In *Proceedings of the International Symposium on Robotics Research*, pages 475–486, 1996.

[27] L. H. Matthies, C. F. Olson, G. Tharp, and S. Laubach. Visual localization methods for Mars rovers using lander, rover, and descent imagery. In *Proceedings of the 4th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 413–418, 1997.

[28] C. F. Olson. Uncertainty estimation in image matching. In progress.

[29] C. F. Olson. Mobile robot self-localization by iconic matching of range maps. In *Proceedings of the International Conference on Advanced Robotics*, pages 447–452, 1997.

[30] C. F. Olson. A probabilistic formulation for Hausdorff matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 150–156, 1998.

[31] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, January 1997.

[32] C. F. Olson and L. H. Matthies. Maximum-likelihood rover localization by matching range maps. In *Proceedings of the International Conference on Robotics and Automation*, pages 272–277, 1998.

[33] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13:471–494, 1966.

[34] W. J. Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance.* Springer-Verlag, 1996.

[35] D. Shirley and J. Matijevic. Mars Pathfinder microrover. *Autonomous Robots*, 2:283–289, 1995.

[36] K. T. Simsarian, T. J. Olson, and N. Nandhakumar. View-invariant regions and mobile robot self-localization. *IEEE Transactions on Robotics and Automation*, 12(5):810–816, October 1996.

[37] F. Stein and G. Medioni. Map-based localization using the panoramic horizon. *IEEE Transactions on Robotics and Automation*, 11(6):892–896, December 1995.

[38] R. Talluri and J. K. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Transactions on Robotics and Automation*, 8(5):573–584, October 1992.

[39] R. Talluri and J. K. Aggarwal. Position estimation techniques for an autonomous mobile robot - A review. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, chapter 4.4, pages 769–801. World Scientific, 1993.

[40] W. B Thompson, T. C. Henderson, T. L. Colvin, L. B. Dick, and C. M. Valiquette. Vision-based localization. In *Proceedings of the DARPA Image Understanding Workshop*, pages 491–498, 1993.

[41] R. Volpe. Navigation results from desert field tests of the Rocky 7 Mars rover prototype. Submitted to *International Journal of Robotics Research.*

[42] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[43] Z. Zhang and O. Faugeras. A 3d world model builder with a mobile robot. *International Journal of Robotics Research*, 11(4):269–285, August 1992.